Supplementary Material for MonoNeRF: Learning a Generalizable Dynamic Radiance Field from Monocular Videos

Fengrui Tian[1] Shaoyi Du[1] Yueqi Duan[2†]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]Department of Electronic Engineering, Tsinghua University
tianfr@stu.xjtu.edu.cn, dushaoyi@gmail.com, duanyueqi@tsinghua.edu.cn

Table 1: Configurations for generalizable dynamic field.

parameter	α_{full}	α_{opt}	α_{corr}	α_{db}	α_{mf}	ϵ
value	1	0.02	4	0.01	1	0.03

1. Supplemental Video

We recommend the readers to watch our supplemental video for more visualization comparisons.

2. Implementation Details

In this section, we present specific implementation details of our model and each experimental setting. The entire model was trained on a NVIDIA A100 GPU with a total batch size of 1024 rays. The learning rate is 0.0005 without decaying. The initial training time is about 3 hours.

2.1. Generalizable Dynamic Field

We trained the generalizable dynamic field in an end-to-end manner. The overall loss function is

$$L = \alpha_{full}L_{full}+\alpha_{opt}L_{opt}+\alpha_{corr}L_{corr}+\alpha_{db}L_{db}+\alpha_{mf}L_{mf}. \quad (1)$$

The hyper-parameter values of all loss functions are listed in Table 1. ϵ is the blending thickness as described in the L_{db} in the paper. We used the Slowonly50 network in SlowFast [3] as the video encoder E_{dy}, which was pretrained on the Kinetics400 [1] dataset, and removed all the temporal pooling layers in the network. We froze the weights of the pretrained model. The temporal features $\boldsymbol{F}_{temp}(\boldsymbol{V})$ are the latent vectors of size 256 extracted by the encoder E_{dy} and fused by W_{dy}, and the spatial features $\boldsymbol{F}_{st}(\boldsymbol{V};\boldsymbol{p})$ of each point were extracted prior to the first 3 spatial pooling layers, which were upsampled using bilinear interpolation, concatenated in the channel dimension and fused with the fully connected layers to form latent vectors of size 256. To incorporate the point feature into NeRF [6] network, we followed pixelNeRF [9] to use multi-layer perceptron (MLP) with residual modulation [5] as our basic block. We employed 4 residual blocks to implement our implicit velocity field, and another 4 residual blocks as the rendering network.

†: Corresponding author.

Figure 1: Qualitative results on the Nerfies [7] dataset .

Table 2: Quantitative results on the Nerfies [7] dataset.

	NeRF [28] + time	DynNeRF [13]	MonoNeRF
PSNR ↑ / LPIPS ↓	23.80 / 0.684	25.80 / 0.671	**27.77 / 0.501**

2.2. Generalizable Static Field

We used ResNet18 [5] as the image encoder E_{st}, which was pretrained on ImageNet [2]. The point features $\boldsymbol{F}_{st}$ in generalizable static field were incorporated prior to 4 pooling layers in ResNet18, which were upsampled and concatenated to form latent vectors of size 256 aligned to each point. We also used MLP with residual modulation as basic architecture for static rendering network. To train the generalizable static field, we used the segmentation mask preprocessed by DynNeRF [4] and optimized the static field by

using the image pixels that belong to static background.

2.3. Novel View Synthesis on Unseen Frames

Novel view synthesis on unseen frames aims to test the generalizable ability of our model on unseen motions in a fixed static scene. We used the first 4 frames to train our generalizable dynamic field, and evaluated the performance on the rest 8 frames for each scene. The training step was set to 40000 in this setting.

2.4. Novel View Synthesis on Unseen Videos

Novel view synthesis on unseen videos aims to test the generalizable ability of our model on novel dynamic scenes. We pretrained our model on Balloon2 scene and finetuned the model on other scenes with the pretrained parameters. The pretraining step on Balloon2 scene was set to 20000. We used the official model [4] trained on Balloon2 scene for a fair comparison. The initial training time is about 3 hours for one scene, and the finetuning time is about 10 minutes.

2.5. Scene Editing

All the scene editing operations are conducted directly on the extracted backbone features without extra training. **Changing background** was conducted by exchanging the extracted image features in static field in our model. **Moving foreground** was implemented by moving the video features in the dynamic field at the corresponding position. **Scaling foreground** was implemented by scaling the video features in the dynamic field. **Duplicating foreground** was conducted by copying the video features to the corresponding position. **Flipping foreground** was applied by flipping the video features. Since the above operations are independent to each other, they can be combined in an arbitrary way.

3. Generalization Results across Datasets

We pretrained our model on the Dynamic Scene dataset [8] and fine-tuned the model on the Nerfies [7] dataset with 500 steps, which contains videos recorded by cellphone cameras. Both Table 2 and Figure 1 show that MonoNeRF presents stronger generalization ability on cellphone videos.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, pages 6201–6210, 2019. 1

[4] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 1, 2

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016. 1

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[7] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 1, 2

[8] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, June 2020. 2

[9] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1